IN THE CLAIMS

1. (Currently Amended) A method, comprising:

receiving a text string having a plurality of characters; and

performing an unanchored search of a database of a stored patterns matching one or more characters of the text string using a state machine, wherein the state machine comprises a ternary content addressable memory (TCAM) and wherein the performing comprises comparing a state of the state machine and one of the plurality of characters with contents of a state field and a character field, respectively, stored in the TCAM, wherein the contents of the state field and the character field stored in the TCAM embody state transitions of the state machine.

2. (Original) The method of claim 1, wherein the state is a next state.

3. (Original) The method of claim 2, further comprising receiving the next state from an associated memory.

4. (Original) The method of claim 1, wherein the state is an idle state.

5. (Original) The method of claim 1, wherein the TCAM implements an Aho-Corasick algorithm.

6. (Original) The method of claim 1, wherein performing the unanchored search comprises:

traversing the state machine with the text string, wherein the state machine is traversed with one of the plurality of characters at a time; and

transitioning a state of the state machine based on a stored next state.

7. (Original) The method of claim 6, further comprising encoding the next state in a lookup table.

8. (Original) The method of claim 7, wherein the plurality of characters comprises valid and invalid characters and wherein encoding comprises encoding the next state in the state machine if a valid character is received in the text string.

9. (Original) The method of claim 8, wherein transitioning further comprises transitioning the state machine to a default state if an invalid character is received in the text string.

10. (Original) The method of claim 8, wherein the transitioning is stopped when an invalid character is received.

11. (Original) The method of claim 1, wherein performing the unanchored search comprises searching for an exact match of one of the stored patterns.

12. (Original) The method of claim 1, wherein performing the unanchored search comprises searching for an inexact match of one of the stored patterns.

13. (Original) The method of claim 1, wherein the TCAM has a first width and the text string has a second width greater than the first width of the TCAM.

14. (Original) The method of claim 1, wherein each of the plurality of characters has a case, and wherein performing the unanchored search further comprises performing the unanchored search insensitive to the case of one or more of the plurality of characters.

15. (Original) The method of claim 14, wherein the text string is encoded in a format having a first plurality of bits, wherein one bit of the first plurality of bits corresponds to the case, wherein the contents of the state field has a second plurality of bits and wherein performing the search insensitive to the case comprises:

masking out the one bit corresponding to the case; and

comparing the first plurality of bits with the second plurality of bits.

16. (Original) The method of claim 15, wherein performing the search insensitive to the case further comprises transforming the characters of the text string from a first code to a second code, the second code having a bit unused in the comparing.

17. (Original) The method of claim 1, wherein the text string has zero or more wildcard characters, zero or more prefix characters preceding the wildcard characters and zero or more suffix characters succeeding the wildcard characters, and wherein performing the unanchored search comprises:

searching the database for a first pattern matching the prefix characters; and

searching the database for a second pattern matching the suffix characters.

18. (Original) The method of claim 17, wherein performing the unanchored search further comprises creating a count that equals a number of the suffix characters plus a number of the wildcard characters.

19. (Previously Presented) The method of claim 1, wherein performing the unanchored search comprises:

comparing, in parallel, N number of the characters with the contents of the state field.

20. (Original) The method of claim 1, wherein the performing further comprises converging all branches of the state machine, for a given stored pattern, to a single next state when a first number of the characters are matched to the contents of a state field to all state transitions of the branches.

21. (Original) The method of claim 20, wherein the single next state is an earlier possible next state for at least one of the branches and wherein the converging comprises transitioning at least one of the branches to the earlier possible next state.

22. (Original) The method of claim 19, further comprising:

storing the characters in a first-in-first-out (FIFO) storage element having a plurality of positions;

positioning a read pointer at a first position; and

adjusting the read pointer to a second position by an amount equal to N minus 1.

23. (Previously Presented) A method, comprising:

receiving a text string having a plurality of characters including a first number of prefix characters, a second number of wildcard characters succeeding the prefix characters, and a third number of suffix characters succeeding the wildcard characters;

performing a first search on a ternary content addressable memory (TCAM) for a first stored pattern matching the prefix characters, wherein the first pattern stored in the TCAM includes state information indicative of a state machine and includes character information indicative of the first pattern; and

performing a second search of the TCAM for a second stored pattern matching the suffix characters, wherein the second pattern stored in the TCAM includes state information indicative of the state machine and includes character information indicative of the second pattern.

24. (Original) The method of claim 23, further comprising creating a count that equals a number of the suffix characters plus a number of the wildcard characters.

25. (Original) The method of claim 23, wherein each of the plurality of characters has a case, and wherein the first and second searches are insensitive to the case.

26. (Original) The method of claim 23, wherein the TCAM has a first width and the text string has a second width greater than the first width.

27. (Original) The method of claim 23, further comprising:

returning a match result when the first stored pattern matches the prefix characters, the second stored pattern matches the suffix characters, and second number of wildcard characters is variable.

28. (Original) The method of claim 23, further comprising:

returning a match result when the first stored pattern matches the prefix characters, the second stored pattern matches the suffix characters, and second number of wildcard characters is fixed.

29. (Original) The method of claim 28, further comprising:

storing a count value that equals a number of the suffix characters plus the fixed second number of the wildcard characters; and

maintaining a count of incoming characters of the text string after receiving the prefix characters; and

returning the match result when the maintained count is equal to the stored count value.

30. (Withdrawn)

31. (Currently Amended) A method, comprising:

receiving a text string having a plurality of characters; and

performing a search of a database of a stored pattern matching one or more characters of the text string using a state machine, wherein the state machine comprises a ternary content addressable memory (TCAM) and wherein the performing comprises comparing a state and one of the plurality of characters with the contents of a state field and a character field, respectively, stored in the TCAM, wherein each of the plurality of characters has a case, ~~and~~ wherein the search is performed insensitive to the case and wherein the contents of the state field and the character field stored in the TCAM embody state transitions of the state machine.

32. (Original) The method of claim 31, wherein the text string is encoded in a format having a first plurality of bits, wherein one bit of the first plurality of bits corresponds to the case, wherein the contents of the state field has a second plurality of bits and wherein performing the search insensitive to the case comprises:

masking out the one bit corresponding to the case; and

comparing the first plurality of bits with the second plurality of bits.

33. (Original) The method of claim 31, wherein performing the search insensitive to the case further comprises transforming the characters of the text string from a first code to a second code, the second code having a bit unused in the comparing.

34-44. (Withdrawn)

45. (Currently Amended) A string search apparatus, comprising:

control circuitry to receive a text string having a plurality of characters; and

a pattern and state database including a ternary content addressable memory (TCAM) coupled to an associated memory, wherein the pattern and state database is operable to perform an unanchored search of the plurality of characters with patterns stored in the TCAM and associated memory by comparing a state of the state machine and one of the plurality of characters with contents of a state field and a character field, respectively, within the patterns stored in the TCAM, wherein the contents of the state field and the character field stored in the TCAM embody state transitions of the state machine.

46. (Original) The string search apparatus of claim 45, further comprising a processor coupled to the pattern and state database.

47. (Original) The string search apparatus of claim 45, wherein the control circuitry comprises:

a first-in-first-out (FIFO) storage element; and

a register coupled to the FIFO storage element and the TCAM.

48. (Original) The apparatus of claim 47, wherein the control circuitry further comprises a rollback circuit coupled to the FIFO storage element.

49. (Original) The apparatus of claim 45, wherein the pattern and state database implements an Aho-Corasick algorithm.